IN THE SPECIFICATION

Please replace the paragraph spanning pages 6 and 7 with the following amended paragraph:

Fig. 1 shows an example of a system configuration according to the present invention in a block diagram. The system of Fig. 1 includes a server computer 101 and client computers 102 to 105. The server computer 101 and the client computers 102 to 105 are connected to a network to communicate various information therebetween. Each client computer can achieve a processing in which the client computer registers a job 106 to the server computer 101, the server computer 101 executes the job 106 in a batch processing manner, and results of the processing are returned to the client computer. In the registration of a job 106, a program executing instruction, and job information 107 which are described in a job control language or script are transferred to the server computer 101.

Please replace the paragraph spanning pages 7 and 8 with the following amended paragraph:

The server computer 101 and the respective client computers 102-105 may use mutually different operating systems. The client computers 102-105 may include various types of clients such as a job launcher client 108 and a www browser client 109. The job registration 107 may be conducted using a predetermined application program interface 111 from a user program 110 of any type. To register a job 106 from a client computer 102 to a server computer 101, there may be used, for example, an operation to input a predetermined command or an operation to instruct the registration using a predetermined graphical user interface (GUI). The job

registered to the server computer 101 is once registered to a queue 112. The client computer 102 can issue an enquiry to the server computer 101 to refer to an execution status of the job 106, to terminate the job 106, or to change the job 106. At the registration of a job 106, the client computer 101 receives a job identifier 113 corresponding to the job 106 from the server computer 101. The job identifier 113 is assigned by the server computer 101. By specifying a job identifier 114, the client computer 102, for example, can issue an enquiry to the server computer 101 to receive a status 115 of the job 106 therefrom. The client computer 102 can display the status on a screen 116 of the client computer 102.

Please replace the paragraph spanning pages 9 and 10 with the following amended paragraph:

The request receiver 211 of the server 208 analyzes the request data stream received from the client 201 to execute processing for each request type of the request data stream. The request type includes "registration", "termination", "change", "reference", and "enquiry". When the request received by the request receiver 211 indicates "registration", the job identifier manager 209 assigns a job identifier thereto and acquires a job identifier stored in the identifier control table 217 of the job identifier storage 210 with a correspondence established between the job identifier and the external identifier. The request receiver 211 adds the job identifier to job information 107, and the queue controller 213 registers the job information 107 to the queue storage 216. The job information 107 is controlled using the queue control table 218. The job information 107 in the request data stream for the registration includes a job name, an execution program name, parameters and

environmental variables necessary for the execution. The job identifier is stored in the job identifier storage 210 and is controlled such that the job identifier includes a unique value in any situation. After the job is registered, the response transmitter 212 returns a response data stream (RES) to the response receiver 206 of the client 201. The response data stream includes the job identifier.

Please replace the paragraph on page 10, lines 5-20, with the following amended paragraph:

The status display section 207 of the client 201 generates a request data stream for "reference". The request data stream includes a job identifier of a job for the reference request. Having received the reference request, the request receiver 211 of the server 208 issues an "enquiry" for an execution status of the job via the queue controller 213 to the execution controller 215. The execution controller 213 acquires an execution status of the job from the queue control table 218 in the queue storage 216 and sends a response data stream via the response transmitter 212 to the response receiver 206 of the client 201. The response data stream includes the job execution status and job information 107. The received response data stream is passed to the status display section 207, which then displays the status of the job on its screen.

Please replace the paragraph spanning pages 10 and 11 with the following amended paragraph:

When the client 201 cannot receive the job identifier due to, for example, a network failure, the client 201 uses an external identifier to issue an enquiry for the

job. For a job of which the job identifier is unknown, the request generator 203 of the client 201 generates a request data stream to send an "enquiry" for the job identifier. The request data stream includes the external job identifier of the job. Having received the enquiry for the job identifier, the request receiver 211 of the server 208 issues an associated request to the job manager 209. The job identifier manager 209 acquires the job identifier of the job from the identifier control table 217 and then the response transmitter 212 sends a response data stream to the response receiver 206 of the client 201. The response data stream includes the job identifier. Having received the job identifier, the response receiver 206 can ask the request generator 203, using the job identifier, to generate a request for the execution status of the job.

Please replace the paragraph on page 11, lines 13-21, with the following amended paragraph:

Fig. 3 shows [[a]]an example format of data communicated between the client 201 and the server 208, as shown in Fig. 2. Fig. 4 shows an example of communication data between the client 201 and the server 208, as shown in Fig. 2. A request data stream includes a header field 301 and a body field 302. The header field 301 includes a general header field 303 and a request header field 304. The body field 302 includes a request body field 305 including request data parameters and the like.

Please replace the paragraph spanning pages 13 and 14 with the following amended paragraph:

Fig. 5 shows a sequence of operation for the client 201 to issue a processing request to the server 208 as shown, for example, in Fig. 2. To register a job to the server 208, the client 201 registers an external identifier within the client 201 (step 501) and issues a job registration request (step 502). For the request, the client 201 sets the external identifier to the request data stream (503). Having received the request data stream, the server 208 analyzes the request (step 504) and registers a job identifier within the server 208 (step 507). From when the client 201 issues the job registration request to when the server 208 conducts the job identifier registration (indicated by an arrow A in Fig. 5), the client 201 can issue an enquiry regarding the job to the server 208 using the external identifier. In the example of Fig. 5, the client 201 issues an enquiry using a request data stream including an external identifier designated by the client 201 (step 505), and the server 208 returns a response data stream including a status of the job to the client 201 (step 506).

Please replace the paragraph on page 14, lines 9-16, with the following amended paragraph:

If the server 208 has already registered the job identifier (step 507), the server 208 receives the enquiry using the external identifier from the client 201 (step 508) and sets the registered job identifier to a response data stream ~~(506)~~(509) and then returns the response data stream to the client 201. Having received the response data stream thereafter, the client 201 can issue an enquiry for the job using the job identifier.

Please replace the paragraph on page 14, lines 17-24, with the following amended paragraph:

For a job status enquiry issued with a job identifier from the client 201 (steps 511, 514, and 517), the server 208 returns a response data stream (512) indicating a queuing status to the client 201 if the job has been registered to the queue (step 510). If the job is in execution (step 513), the server 208 sends a response data stream (515) indicating that the job is in execution to the client 201.

Please replace the paragraph on page 14, lines 25-28, with the following amended paragraph:

After the job is completely processed and the server 208 sends results of the processing to the client 201 (step 516), the server 208 sends a response data stream (518) indicating a status of termination to the client 201.

Please replace the paragraph on page 15, lines 1-3, with the following amended paragraph:

In this example, the client 201 having received the job status displays the results of processing on a screen (step 519).

Please replace the paragraph on page 15, lines 4-11, with the following amended paragraph:

During a period of time in which information regarding the job is being kept in the server (indicated by an arrow B in Fig. 5) after the server 208 registers the job identifier, the client 201 can issue a request for the job using the job identifier to the

server 208. In the period indicated by the arrow B, the client 201 can also use the external identifier for the enquiry.

Please replace the paragraph on page 15, lines 12-14, with the following amended paragraph:

Fig. 6 shows an example in which a response data stream is sent to the client 201 when the server 208 is in process.

Please replace the paragraph on page 15, lines 15-27, with the following amended paragraph:

At a job registration request (step 602), the client 201 can set a mode in which at start or termination of each processing, the server 208 informs the client 201 of a status of the job, using a response data stream. This becomes possible when the client 201 sets information, for example, of a client 201 to which the status is to be informed and a port number associated with the client 201, to job information of a request data stream (603) at the job registration request. By appropriately setting the job information, the client 201 can also select a condition to inform the client 201 of the job status, the condition indicating a status of processing in the server 208.

Please replace the paragraph spanning pages 15 and 16 with the following amended paragraph:

In the example of Fig. 6, the client 201 sets an external identifier (step 601) to send the external identifier using a request data stream (step 603) of a registration request (step 602); the server 208 then analyzes the request (step 604); moreover, a

status of the job is informed to a requester of the job at registration of a job identifier (step 605), at registration of a queue (step 607), at execution of processing (step 609), and at delivery of results of processing (steps 606, 608, 610, 611, 612 and 613). Therefore, necessary information of a status of the job can be obtained as a default value without delivering a request data stream from the client 201 to the server 208 in this example, and hence the network load can be reduced. Additionally, when necessary, using the external identifier or the job identifier, an enquiry to the server 208 can be issued for necessary information of the job.

Please replace the paragraph spanning pages 17 and 18 with the following amended paragraph:

Fig. 9 is an embodiment in which the present invention is applied to an electronic commerce system. In an operation to order items from a customer computer 902 to an order receiving computer 901, the customer computer 902 delivers order information 913 via an ~~internet~~Internet 903 to the order receiving computer 901. The order information 913 includes unique information [[913]]914 such as a customer registration number, an order identifier 915 to uniquely identify each order of the customer from the customer computer, and order contents information 916 specifying an item name of the ordered item, the number of items, specifications of the item, and the like. The order receiving computer 901 conducts ~~operations~~an order receiving processing (904), including, for example, ~~receives~~ receiving the order (step 905), ~~registers~~ registering the order (906), ~~and confirms~~confirming the stock (step 907), creating bill (908), indicating delivery (909-910), replenishing backlog (911), and indicating order (912). However, since a

system failure may occur due to communication quality of the ~~internet~~Internet and/or a concentrated load of a large number of accesses to the order receiving computer 901, the order information 913 transmitted from the customer is not necessarily registered in the order receiving computer 901. Therefore, by delivering an identifier including the unique information 914 and the order identifier information 915 delivered together with the order information 913 to the order receiving computer 901, it is possible to prevent the problem in which an item is twice ordered or an item ordered cannot be delivered.